

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Державний університет інформаційно-комунікаційних технологій
Кафедра Технології цифрового розвитку

МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ КУРСОВОГО ПРОЄКТУ З ООП С#

для студентів спеціальності 121 Інженерія програмного забезпечення
першого (бакалаврського) рівня

Київ 2025

Розробники:

Гавор Артур Станіславович, викладач кафедри Технології цифрового розвитку

Методичні вказівки до курсового проєкту “Об'єктно-орієнтоване програмування С#” з дисципліни “ Об'єктно-орієнтоване програмування С# ” затверджені на засіданні кафедри інженерія програмного забезпечення протокол від 29 серпня 2025р. № 1

Завідувач кафедри Технологій цифрового розвитку

В.В.Жебка

“ _____ ” _____ 20__ року

Методичні вказівки до виконання курсового проєкту з С# для студентів спеціальності 121 Інженерія програмного забезпечення першого (бакалаврського) рівня.

Методичні вказівки містять загальні положення щодо мети, змісту та організації виконання курсового проєкту з С#, детальний опис всіх структурних елементів роботи та вимоги до оформлення. Подається порядок і процедура допуску до захисту та самого захисту й критерії оцінювання. У додатках наведено зразки документів, які використовуються при підготовці курсового проєкту. Є керівним документом для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 121 Інженерія програмного забезпечення та їх керівників при написанні курсового проєкту з С#.

Зміст

Передмова	4
1. Підготовчий етап виконання курсового проєкту	5
1.1. Організація взаємодії керівника та студента	5
1.2. Вибір теми та окреслення актуальності	6
1.3. Окреслення орієнтовного плану проєкту та термінів виконання	7
2. Основний та оформлювальний етап виконання курсового проєкту	8
2.1. Компоненти курсового проєкту, що представляються на захист	8
2.2. Вимоги до структури та змісту текстового документу	9
2.3. Вимоги до оформлення курсового проєкту	13
2.4. Вимоги до структури та оформлення презентації	19
2.5. Вимоги до відеоролика-представлення роботи	21
2.6. Підготовка до виступу-представлення наукової роботи	21
2.7. Апробація отриманих результатів	22
2.8. Академічна доброчесність при написанні курсового проєкту	23
3. Порядок захисту й оцінювання курсового проєкту	23
3.1. Процедура захисту	23
3.2. Критерії оцінювання курсового проєкту	25
Список використаних джерел	26
ДОДАТКИ	28

Передмова

Відповідно до Положення про організацію навчального процесу у закладах вищої освіти України, курсовий проєкт спрямований на закріплення, поглиблення та узагальнення знань, отриманих здобувачами вищої освіти під час навчання, а також на їх практичне застосування для комплексного вирішення фахових завдань. Написання та захист курсового проєкту є важливим етапом підготовки здобувачів вищої освіти до виконання більш складних дослідницьких і проєктних завдань, зокрема бакалаврських і магістерських кваліфікаційних робіт.

Курсовий проєкт з дисципліни «Об'єктно-орієнтоване програмування (C#)» є обов'язковим елементом освітньо-професійної програми «Технології цифрового розвитку» для здобуття ступеня бакалавра за спеціальністю 121 «Інженерія програмного забезпечення». Підготовка фахівців зазначеної спеціальності передбачає обов'язкову практичну складову, частково реалізовану через виконання курсових проєктів.

Процес організації, написання, оформлення та захисту курсових проєктів регламентується Положенням про випускні кваліфікаційні роботи (проєкти) Державного університету інформаційно-комунікаційних технологій. Курсовий проєкт з ООП С# є самостійною навчально-науковою роботою здобувача вищої освіти, що виконується в межах дисципліни «ООП С#» та є підсумком опанування відповідного навчального матеріалу.

Курсовий проєкт має представляти завершену розробку прикладної задачі професійного спрямування та відповідати таким вимогам:

- бути актуальним, містити елементи новизни та виконуватися з урахуванням сучасних досягнень науки і техніки;
- орієнтуватися на розв'язання практичних завдань, пов'язаних із майбутньою професійною діяльністю;
- заохочувати здобувачів вищої освіти до творчого пошуку ефективних рішень поставлених задач;
- передбачати опрацювання, вивчення та аналіз спеціалізованої наукової й методичної літератури;
- включати вибір оптимальних рішень на основі аналізу алгоритмів, вимог до системи та застосування сучасних методів розробки програмного забезпечення.

Метою курсового проєкту є систематизація, закріплення та поглиблення теоретичних і практичних знань, здобутих у процесі вивчення дисципліни «ООП С#», а також розвиток умінь застосовувати ці знання для розв'язання прикладних задач у сфері інженерії програмного забезпечення.

Основні завдання курсового проєкту з ООП С#:

- поглиблене вивчення принципів об'єктно-орієнтованого проєктування та програмування, мови С# і платформи .NET, а також формування здатності застосовувати відповідні знання під час виконання професійних завдань;

- набуття практичних навичок розробки програмного забезпечення із використанням моделей і методологій життєвого циклу програмного забезпечення та актуальних сучасних практик розробки ПЗ;
- формування практичних умінь щодо аналізу прикладної задачі, проєктування архітектури програмного рішення та реалізації програмних компонентів;
- реалізація у вигляді програмного продукту одного або кількох взаємопов'язаних функціональних компонентів з урахуванням принципів модульності та масштабованості;
- застосування основних інструментів і фреймворків платформи .NET для створення серверних та/або настільних застосунків відповідно до обраної архітектури;
- набуття практичних навичок керування проєктом, тестування програмного забезпечення та документування розробленого продукту з урахуванням вимог до якості програмного забезпечення.

1. Підготовчий етап виконання курсового проєкту

1.1. Організація взаємодії керівника та студента

Керівником курсового проєкту з ООП С# призначається науково-педагогічний працівник кафедри, закріплений за викладанням дисципліни «ООП С#» або суміжних дисциплін програмування. Здобувач вищої освіти має право самостійно обрати керівника з переліку викладачів, запропонованого кафедрою. Перед вибором керівника рекомендується ознайомитися з його науковими інтересами та публікаціями.

Робота над курсовим проєктом спрямована на розвиток навичок постановки та розв'язання прикладних задач, самостійної роботи з літературними джерелами, вибору й освоєння необхідного інструментарію. Курсовий проєкт виконується здобувачем самостійно з можливістю отримання консультацій від керівника, який надає методичну підтримку та допомагає визначити оптимальні підходи до реалізації поставленої задачі.

Основні обов'язки керівника курсового проєкту:

- допомога у виборі та формулюванні теми курсового проєкту з урахуванням наукових і практичних інтересів здобувача;
- сприяння у складанні плану роботи та визначенні термінів виконання;
- надання консультацій щодо структури проєкту та його змістового наповнення;
- рекомендації з пошуку, відбору та аналізу джерел інформації;
- консультації щодо реалізації функціоналу, вибору оптимальних алгоритмів і технологій розробки;
- контроль виконання основних етапів роботи відповідно до затвердженого плану;

- перевірка дотримання академічної доброчесності;
- підготовка здобувача до захисту курсового проєкту;
- оцінювання якості роботи та прийняття рішення щодо допуску до захисту.

Основні обов'язки студента під час написання курсового проєкту

включають:

- своєчасне звернення до керівника для отримання консультацій щодо вибору та формулювання теми проєкту;
- розроблення плану виконання курсового проєкту з визначенням основних етапів і термінів виконання;
- дотримання календарного графіка та виконання всіх етапів проєкту у встановлені строки;
- забезпечення дотримання принципів академічної доброчесності під час виконання роботи;
- регулярне подання керівнику проміжних результатів та внесення змін і доопрацювань відповідно до його зауважень і рекомендацій.

Курсовий проєкт виконується згідно із затвердженим календарним планом. Кафедра формує графік консультацій керівників, у якому зазначено час і місце їх проведення. Консультації з керівником проводяться відповідно до затвердженого графіка (рекомендовано – не рідше одного разу на тиждень).

1.2. Вибір теми та окреслення актуальності

Тематика курсового проєкту має відповідати професійним завданням, визначеним освітньо-професійною програмою та освітньо-кваліфікаційною характеристикою за напрямом підготовки. Курсовий проєкт з дисципліни «ООП С#» повинен бути актуальним, відповідати змісту дисципліни та сучасним тенденціям розвитку інженерії програмного забезпечення, зокрема технологіям платформи .NET.

Назва проєкту повинна бути лаконічною, зрозумілою та відображати його мету й призначення. Курсовий проєкт з ООП С# має бути спрямований на розв'язання однієї або кількох взаємопов'язаних задач прикладного характеру та обов'язково передбачати програмну реалізацію запропонованого рішення. Поставлена задача повинна бути повністю виконана, а проєкт – завершений та придатний до демонстрації під час захисту.

Основним результатом виконання курсового проєкту з ООП С# є створення програмного продукту, розробленого мовою С# на платформі .NET із застосуванням принципів об'єктно-орієнтованого програмування.

Перелік тем курсових проєктів формується кафедрою та оновлюється щороку. Здобувачі вищої освіти можуть обрати тему зі запропонованого переліку або запропонувати власну тему, обґрунтувавши її актуальність і доцільність. У випадку, якщо здобувач у встановлений термін не визначився з темою курсового проєкту або керівником, відповідні рішення приймаються кафедрою на засіданні.

Тематика курсових проєктів оголошується на початку семестру, затверджується рішенням кафедри та оприлюднюється у встановленому порядку не пізніше ніж за три місяці до дати захисту. Уточнення формулювання теми допускається лише за погодженням керівника та за рішенням кафедри, але не пізніше ніж за один місяць до захисту.

Здобувач не має права самостійно змінювати затверджену тему курсового проєкту без погодження керівника та кафедри. Виконання різними здобувачами курсових проєктів на одну і ту саму тему не допускається. Разом з тим можлива командна реалізація одного проєкту за умови чіткого розподілу ролей і відповідальності (за окремі модулі або підсистеми), а також наявності індивідуальних результатів у кожного учасника.

Основними критеріями вибору теми курсового проєкту з ООП С# є:

- актуальність, наявність елементів новизни та перспективність обраної теми;
- ступінь дослідженості теми в науковій та прикладній літературі;
- наявність доступних джерел інформації та матеріалів, достатніх для розкриття теми;
- можливість реалізації теми в межах технічних можливостей та напряму роботи кафедри;
- науково-практичні інтереси здобувача вищої освіти.

1.3. Окреслення орієнтовного плану проєкту та термінів виконання

Після визначення теми курсового проєкту здобувач вищої освіти (студент) проходить першу настановчу консультацію у керівника курсового проєкту. Під час консультації уточнюються загальні вимоги до роботи, порядок виконання курсового проєкту, структура пояснювальної записки, етапи реалізації програмного продукту, терміни виконання окремих розділів, перелік рекомендованих джерел, а також підхід до проведення аналізу предметної області та проєктування програмного рішення.

Таблиця 1.

Календарний план написання курсового проєкту

№	Назва етапу виконання курсового проєкту	Терміни виконання етапів проєкту
1	Вибір теми курсового проєкту. Подання заяви про обрання теми. Затвердження теми.	2 тиждень семестру
2	Затвердження графіка виконання роботи.	4 тиждень семестру
3	Добір літератури та первинне ознайомлення з джерелами. Формування бібліографії.	4–5 тиждень семестру
4	Формування плану курсового проєкту.	5 тиждень семестру

Продовження таблиці 1.

Календарний план написання курсового проєкту

5	Опис теоретичних аспектів дослідження. Аналіз літератури. Написання першого розділу курсового проєкту.	5–6 тиждень семестру
6	Проектування та розробка програмного продукту мовою C# на платформі .NET.	7–13 тиждень семестру
7	Тестування та налагодження розробленого програмного продукту.	13 тиждень семестру
8	Опис технології розробки, архітектури та реалізації програмного продукту. Написання другого розділу курсового проєкту.	14 тиждень семестру
9	Оформлення курсового проєкту відповідно до встановлених вимог.	4–14 тиждень семестру
10	Подання курсового проєкту керівнику. Підготовка презентації. Отримання допуску до захисту.	14 тиждень семестру
11	Захист курсового проєкту перед комісією.	15 тиждень семестру

2. Основний та оформлювальний етап виконання курсового проєкту

2.1. Компоненти курсового проєкту, що представляються на захист

Перелік компонентів, які подаються на кафедру перед захистом курсового проєкту:

- лектронний варіант текстової частини курсового проєкту у форматі .doc або .docx;
- електронний варіант програмної розробки у вигляді архіву або посилання на репозиторій, що містить вихідний код, файл рішення (.sln), файл проєкту (.csproj), а також інструкцію запуску (README) із зазначенням версії .NET та необхідних залежностей (за наявності);
- репозиторій системи контролю версій Git (GitHub, GitLab, Bitbucket або інший), що містить історію розробки, коміти та актуальну версію проєкту, а також файл README.md з інструкцією щодо запуску та використання програмного продукту;
- друкований примірник текстової частини курсового проєкту з резолюцією «До захисту», датою та підписом керівника на титульному аркуші;
- відеоролик-презентація програмної розробки (демонстрація роботи застосунку та основних сценаріїв).

2.2. Вимоги до структури та змісту текстового документу

Орієнтовний обсяг курсових проєктів: текст роботи – 25-30 сторінок; наявність обов'язкового ілюстративного матеріалу (скріншоти, UML-діаграми тощо).

Обов'язковими структурними елементами є:

Вступна частина:

- титульний аркуш;
- зміст;
- перелік умовних позначень (за потребою);
- вступ.

Основна частина:

- розділи;
- висновки;
- список використаних джерел.

Додатки:

- основні документи програмної документації;
- інструкція користувачу.

Титульний аркуш. Титульний аркуш оформляється згідно зразку наведеного у додатку А. Тема роботи повинна бути ідентичною темі, затвердженій на кафедрі. У випадку невідповідності теми, робота до захисту не допускається.

Зміст. Зміст текстової частини курсового проєкту подають на початку роботи. Зміст повинен послідовно містити назви всіх структурних елементів роботи, окрім титульного листа та самого змісту, а також посилання на номери сторінок, на яких починається даний структурний елемент. Шаблон та зразок оформлення змісту подано в додатках Б, В.

Перелік умовних позначень. Перелік умовних позначень є не обов'язковою структурною частиною роботи. Даний перелік складається, якщо в роботі зустрічаються маловідомі скорочення, специфічні терміни або аббревіатури. Перелік друкується двома колонками – в першій колонці подається термін, в другій його детальне пояснення. Якщо в роботі термін зустрічається не більше 3 разів, то його тлумачення не подається в переліку умовних позначень, а розшифровується в самому тексті роботи при першому його згадуванні.

Вступ. Вступ до курсового проєкту повинен мати наступну структуру:

- актуальність теми;
- мета роботи;
- завдання;
- об'єкт дослідження;
- предмет дослідження;
- апробація результатів роботи (у випадку наявності);
- публікації (у випадку наявності).

Обсяг вступу не повинен перевищувати 2-3 сторінки.

Актуальність теми. Актуальність і доцільність роботи для розвитку галузі комп'ютерних наук обґрунтовується через критичний аналіз та порівняння з відомими розв'язками проблеми (наукової задачі). Опис актуальності має бути лаконічним, складатися з кількох речень, чітко окреслити суть проблеми або наукового завдання. Якщо робота є частиною досліджень кафедри (відділу, організації тощо), необхідно коротко вказати зв'язок вибраного напрямку з планами організації, галузевими чи державними програмами. Обов'язково слід зазначити особистий внесок автора в розв'язання цих науково-дослідних завдань.

Мета та завдання дослідження. Визначають мету роботи та конкретні завдання, які необхідно вирішити для досягнення цієї мети. Мета є кінцевим результатом, якого автор прагне досягти в ході дослідження, і повинна відповідати темі роботи. Завдання – це конкретні кроки та методи, що допоможуть досягти поставленої мети. Перелік завдань має бути узгоджений зі змістом роботи.

Об'єкт дослідження – це процес чи явище, яке спричиняє проблему.

Предмет дослідження – це частина об'єкта, яка є безпосереднім об'єктом вивчення. Об'єкт і предмет взаємопов'язані як загальне і часткове. З об'єкта виділяється його частина, яка стає предметом дослідження, на який спрямована увага автора, оскільки саме від нього залежить формулювання назви роботи.

Апробація результатів роботи. Якщо автор брав участь у наукових конференціях, семінарах або засіданнях наукового гуртка, де були оголошені результати його роботи, і має підтвердження цього (опубліковані тези, статті або програма конференції за темою роботи), слід зазначити їх назву, рік та місце проведення.

Публікації. Якщо автор має публікації, то потрібно подати їх перелік. Вказати назву публікації, видання, дату публікації.

Основна частина. Текстова частина курсового проєкту складається з двох розділів. Розділи повинні мати бібліографічні посилання на джерела, що дозволяє аналізувати власні дані та порівнювати їх з даними інших джерел.

У першому розділі зазвичай описуються теоретичні дослідження, пов'язані з темою курсового проєкту, на основі первинних джерел. Подається огляд існуючих підходів до розв'язання поставленої проблеми, аналіз алгоритмів, що застосовуються для її вирішення, та порівняльна характеристика цих підходів. Важливе значення при написанні цього розділу має правильне трактування термінів, їх точність і науковість. Використовувані терміни повинні бути загальноновживаними або наводитися з посиланням на їх авторів. Останнім підрозділом першого розділу має бути «Огляд та аналіз аналогічних програмних розробок». У цьому пункті потрібно здійснити аналіз існуючих програмних продуктів, що виконують подібні функції (2-3 приклади), вказавши їх переваги і недоліки, а також можливість використання цих аналогів для розв'язання конкретної задачі.

При проведенні порівняльного аналізу аналогічних програмних розробок, для кожного аналогу програмного забезпечення студент повинен визначити наступні характеристики:

- назва;
- розробник (дистриб'ютор);
- архітектура (client-server, MVC, мікросервісна, монолітна);
- мова реалізації;
- перелік функцій, характеристик (не менше 5);
- аналіз переваг та недоліків ПЗ;
- джерело інформації (веб-сайт, документація).

Обов'язковим є використання скріншотів проаналізованих програмних продуктів, що відображають їх роботу (1–2 для кожного програмного засобу).

На основі результатів аналізу, поданого в даному підрозділі, мають бути сформовані та вказані вимоги до програмного продукту, який реалізує мету та завдання курсового проєкту.

Назва першого розділу формується згідно текстового наповнення першого розділу і повинна відповідати темі курсового проєкту.

Другий розділ містить опис процесу розробки програмного продукту. Назва другого розділу повинна бути співзвучна з назвою курсового проєкту. Другий розділ обов'язково повинен мати наступні підрозділи:

2.1. Постановка задачі, призначення та вимоги до програмного засобу *«назва програмної розробки»*

2.2. Загальний опис проєкту

2.3. Обґрунтування вибору інструментальних засобів розробки *«назва програмної розробки»*

2.4. Особливості програмної реалізації *«назва програмної розробки»*

2.5. Тестування та налагодження програмного засобу *«назва програмної розробки»*

2.6. Рекомендації щодо використання та впровадження програмного засобу *«назва програмної розробки»*

Дані підрозділи можуть бути переформульовані згідно теми курсового проєкту, розширені і доповнені підпунктами, але при цьому структурний зміст порядку оформлення не повинен бути порушеним.

У підрозділі **«Постановка задачі, призначення та вимоги до програмного засобу...»** повинні бути чітко сформульовані завдання курсового проєкту, описано його призначення, вимоги та наведено перелік функціональних характеристик програмного продукту, що розробляється.

Результатом формування цього підрозділу має стати технічне завдання (зразок оформлення технічного завдання подано в додатку Д).

У підрозділі **«Загальний опис проєкту»** необхідно розробити загальну модель процесів, які відбуваються під час вирішення задачі в межах проєкту. Це включає опис основних принципів програмної реалізації, визначення необхідних структур даних та обґрунтування вибору алгоритмів, які будуть використовуватись для реалізації задачі. Доцільно але не обов'язково привести діаграми класів, послідовності, активностей, компонентів. У межах реалізації

програмного продукту обов'язково мають бути продемонстровані основні принципи об'єктно-орієнтованого програмування (інкапсуляція, абстракція, поліморфізм, спадкування та/або композиція), а також взаємодія класів і відповідність програмної реалізації обраній архітектурі. Особливу увагу слід приділити обґрунтуванню вибору алгоритмів, зокрема, їх ефективності в умовах конкретної проблеми. Крім того, потрібно описати загальний алгоритм роботи програми на рівні логіки, не прив'язуючи його до конкретного середовища та мови програмування. Алгоритми та програмні рішення мають бути реалізовані мовою програмування C# з урахуванням можливостей платформи .NET (типізація, колекції, винятки, LINQ, асинхронність – за потреби), забезпечуючи коректність, ефективність та читабельність коду.

У підрозділі **«Обґрунтування вибору інструментальних засобів розробки...»** необхідно надати детальний опис технологій, середовища та мови програмування, обраних для реалізації поставленої задачі. Також слід провести аналіз доступних інструментальних засобів та обґрунтувати їх вибір.

Підрозділ **«Особливості програмної реалізації...»** повинен містити детальний опис ключових аспектів реалізації алгоритмів, основних компонентів програмного рішення, його модулів та всіх створених функцій. Код програми, наведений у цьому розділі, має супроводжуватися коментарями. У рамках цього підрозділу також необхідно підготувати керівництво користувача, яке оформлюється у вигляді додатку (зразок наведено в додатку Ж).

Підрозділ **«Організація тестування та налагодження програмного засобу...»** повинен містити результати тестування розробленого програмного продукту, виявлені недоліки та рекомендації щодо їх усунення.

У підрозділі **«Рекомендації щодо використання та впровадження програмного засобу...»** необхідно обґрунтувати можливість практичного застосування створеного програмного продукту, вказати умови його експлуатації, за яких забезпечується коректна робота, а також описати необхідні технічні та програмні вимоги для його функціонування.

Висновки повинні містити чіткий виклад найважливіших отриманих результатів, а також рекомендації та пропозиції щодо подальшого дослідження цієї теми. Висновки оформлюються у вигляді окремих, лаконічних положень або рекомендацій, що підсумовують виконану роботу. Важливо, щоб вони відповідали поставленим завданням, визначеним у вступі. У висновках слід зазначити не лише досягнуті позитивні результати дослідження, а й можливі недоліки чи труднощі, а також шляхи їх усунення.

Список використаних джерел. З кількох можливих способів упорядкування матеріалу у списку (за алфавітом, за порядком згадування, за видом джерела). Кількість використаних джерел повинна бути не менше 25. Бібліографічний опис джерел та літератури складають відповідно до чинних стандартів з бібліотечної або видавничої справи (Інформація та документація. Бібліографічне посилання: Загальні положення та правила складання: ДСТУ 8302:2015). Якщо список формується за абеткою і містить посилання на літературу та джерела, записані як кириличними літерами, так і латинськими, то спочатку потрібно подавати ті, які записані кирилицею, а потім латиницею.

Додатки. До обов'язкових додатків відносяться «Технічне завдання» та «Керівництво користувача». За потреби до додатків можуть бути включені інші допоміжні матеріали, що сприяють повнішому розумінню роботи, такі як додаткові ілюстрації, великі фрагменти коду програми (обов'язково з коментарями) тощо. Кожен додаток має бути згаданий у тексті роботи з відповідним посиланням.

2.3. Вимоги до оформлення курсового проєкту

Загальні вимоги до оформлення курсового проєкту. Текст курсового проєкту виконують на аркушах білого паперу формату А4 (210х297 мм).

Основний текст роботи друкують з одного боку аркуша через 1,5 комп'ютерних інтервали шрифтом Times New Roman, розмір шрифту – 14. Абзацний відступ – 1,25 см. Інтервал перед і після абзацу – 0 пунктів. Виділення тексту напівжирним, курсивом, підкресленим або іншим шрифтом, окрім Times New Roman, розмір 14, не допускається, окрім випадків, зазначених нижче (заголовки, окремі слова вступу).

Текст необхідно друкувати, залишаючи береги таких розмірів: лівий – 30 мм, верхній, нижній – 20 мм, правий – 10 мм.

У тексті документа необхідно дотримуватися рівномірної щільності, контрастності та чіткості зображення.

Роботу подають до захисту у переплетеному вигляді.

Курсовий проєкт виконують українською мовою. Текст курсового проєкту повинен бути стислий, точний та логічно послідовний. Текст викладають, дотримуючись норм чинного українського правопису (зі змінами і доповненнями 2019 року), використовуючи стиль ділового мовлення, необхідний для службових документів. Потрібно користуватися усталеною лексикою, наявною в академічних словниках, дотримуватися прийнятої наукової термінології, умовних символів та скорочень, уникати діалектизмів, засобів художнього стилю, нових запозичень тощо. Не можна вживати займенник «я», необхідно вживати займенник «ми».

За необхідності заключення тексту в лапки слід користуватись одним стилем символу по всьому документу.

Формування змісту. Зміст створюється виключно засобами автоматичного генерування змісту текстового процесора і повинен містити посилання на структурні елементи курсового проєкту, включаючи номери сторінок.

У змісті заголовки розділів подаються великими буквами, а підрозділів, пунктів – маленькими з першої великої; у змісті не виділяти заголовки напівжирним.

Заголовки структурних частин «ЗМІСТ», «ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ», «ВСТУП», «РОЗДІЛ», «ВИСНОВКИ», «СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ», «ДОДАТКИ», «Додаток Х» (заголовки першого рівня) у змісті відображати без відступу від лівого берега сторінки, заголовки другого рівня з відступом 0,75 см від лівого берега сторінки, заголовки третього рівня – з відступом 1,5 см від лівого берега сторінки.

Оформлення розділів (підрозділів). Текст основної частини поділяють на розділи, підрозділи, пункти і підпункти. Заголовки структурних частин «ЗМІСТ», «ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ», «ВСТУП», «РОЗДІЛ», «ВИСНОВКИ», «СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ», «ДОДАТКИ» друкують великими літерами симетрично до тексту. Кожну структурну частину роботи починають з нової сторінки, крім назв підрозділів і пунктів у межах розділу. Після заголовків першого рівня «ЗМІСТ», «ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ», «ВСТУП», «РОЗДІЛ», «ВИСНОВКИ», «СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ», «ДОДАТКИ» пропускається один рядок (один абзац, шрифт Times New Roman, розмір шрифту – 14, міжрядковий інтервал – 1,5, інтервал перед і після абзацу – 0).

Назви розділів друкують великими літерами симетрично до тексту. Розділи нумерують арабськими цифрами. Після номера розділу крапка не ставиться. З нового рядка друкують назву розділу великими літерами симетрично до тексту без крапки в кінці.

Відстань між заголовком і подальшим чи попереднім текстом має бути один рядок: шрифт Times New Roman, розмір шрифту – 14, міжрядковий інтервал – 1,5.

Відстань між основами рядків заголовку, а також між двома заголовками приймають такою, як у тексті.

Заголовки підрозділів (заголовки другого рівня) друкують маленькими літерами (крім першої великої) з абзацу, абзацний відступ – 1,25 см (текст заголовку повинен відступати від номера заголовку на 0,5 см) шрифтом Times New Roman, розмір шрифту – 14, напівжирний, міжрядковий інтервал – 1,5, інтервал перед і після абзацу – 0. Крапку в кінці заголовка не ставлять.

Заголовки пунктів (заголовки третього рівня) друкують маленькими літерами (крім першої великої) з абзацу, абзацний відступ – 1,75 см, шрифтом Times New Roman, розмір шрифту – 14, напівжирний, міжрядковий інтервал – 1,5, інтервал перед і після абзацу – 0. Крапку в кінці заголовка не ставлять.

Перенесення слів у заголовках не допускається. Якщо заголовок складається з кількох рядків, другий і наступні рядки заголовку другого рівня друкують з відступом у 1,25 см від лівого берега сторінки; другий і наступні рядки заголовку третього рівня друкують з відступом у 1,75 см від лівого берега сторінки. Якщо заголовок складається з кількох речень, їх розділяють крапкою.

Назви підрозділів і пунктів друкуються в межах розділу, а не починаються з нової сторінки. Не допускається розташування назв підрозділів, пунктів у нижній частині сторінки, якщо після них міститься лише один рядок тексту. В такому випадку такі окремі заголовки примусово переносяться на наступну сторінку.

При нумерації заголовків другого та наступного рівнів відступ від номера заголовку до назви заголовку повинен становити 0,5 см.

У вступі лише слова «Актуальність теми», «Мета роботи», «Завдання», «Об'єкт дослідження», «Предмет дослідження», «Апробація результатів роботи», «Публікації» виділяють напівжирним шрифтом.

Слово «ВИСНОВКИ» друкують великими літерами симетрично до тексту шрифтом Times New Roman, розмір шрифту – 14, напівжирний. Після слова «ВИСНОВКИ» пропускається один рядок (один абзац, шрифт Times New Roman, розмір шрифту – 14, міжрядковий інтервал – 1,5, інтервал перед і після абзацу – 0).

Слова «СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ», друкують великими літерами симетрично до тексту шрифтом Times New Roman, розмір шрифту – 14, напівжирний. Після слів «СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ» пропускається один рядок (один абзац, шрифт Times New Roman, розмір шрифту – 14, міжрядковий інтервал – 1,5, інтервал перед і після абзацу – 0).

Додатки слід позначати послідовно великими літерами української абетки, за винятком літер Г, Є, З, І, Ї, Й, О, Ч, Ь, наприклад: Додаток А, Додаток Б і т.д. Якщо у тексті один додаток, то він позначається як Додаток А.

Додаток повинен мати заголовок, надрукований вгорі малими літерами з першої великої симетрично відносно тексту сторінки. Посередині рядка над заголовком малими літерами з першої великої повинно бути надруковано слово «Додаток __» і велика літера, що нумерує додаток.

Після назви додатку пропускається один рядок (один абзац, шрифт Times New Roman, розмір шрифту – 14, міжрядковий інтервал – 1,5, інтервал перед і після абзацу – 0).

Нумерація. Нумерацію сторінок, розділів, підрозділів, пунктів, підпунктів, малюнків, таблиць, формул подають арабськими цифрами без знаку №.

Сторінки роботи слід нумерувати арабськими цифрами, дотримуючись наскрізної нумерації впродовж усього тексту роботи. Номер сторінки проставляють у правому верхньому куті сторінки без крапки в кінці.

Першою сторінкою випускної кваліфікаційної роботи є титульний аркуш, який включають до загальної нумерації сторінок. На титульному аркуші номер сторінки не ставлять.

Такі структурні частини курсового проєкту як «ЗМІСТ», «ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ», «ВСТУП», «РОЗДІЛ», «ВИСНОВКИ», «СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ», «ДОДАТКИ» не мають порядкового номера. Інші розділи, підрозділи, пункти, підпункти нумерують арабськими цифрами. Номер розділу ставлять після слова «РОЗДІЛ», після номера крапку не ставлять, потім з нового рядка друкують заголовок розділу. Підрозділи повинні мати порядкову нумерацію в межах кожного розділу. Номер підрозділу складається з номера розділу й порядкового номера підрозділу, відокремленого крапкою. Після номера підрозділу (пункту, підпункту) ставлять крапку (наприклад: 1.1.; 1.3.; 1.2.4.). Пункти нумерують у межах кожного підрозділу. Номер пункту складається з порядкових номерів розділу, підрозділу, пункту, між якими ставлять крапку. У кінці номера повинна стояти крапка, наприклад: «1.3.2.» (другий пункт третього підрозділу першого розділу). Потім у тому ж рядку наводять заголовок пункту. Всі аркуші, на яких розміщені згадані структурні

частини, нумерують звичайно. Рисунки і таблиці, розміщені на окремих сторінках, включають до загальної нумерації сторінок документа.

Вимоги до оформлення ілюстрацій, таблиць, формул. В роботі обов'язкова наявність ілюстративного матеріалу: рисунки, графіки, схеми, діаграми.

Ілюстрації слід розташовувати безпосередньо після тексту, у якому вони згадуються вперше, або на наступній сторінці. На всі ілюстрації повинні бути посилання у тексті. Якщо ілюстрації створені не автором роботи, необхідно зробити певні посилання, дотримуючись вимог чинного законодавства щодо авторських прав.

Креслення, рисунки, графіки, схеми, діаграми, які містяться у тексті, повинні відповідати вимогам державних стандартів.

Ілюстрації повинні мати назву, яку розміщують під зображенням. За необхідності під ілюстрацією розміщують пояснювальні дані (напис під рисунком). Ілюстрації позначають словом «Рисунок...», яке разом із назвою ілюстрації розташовують після пояснювальних даних. Наприклад: «Рисунок 3.1 - Схема розміщення» або «Рисунок 3.1 - Діаграма стану...».

Посилання на ілюстрації роботи вказують порядковим номером ілюстрації, наприклад, «Рис. 1.2».

Ілюстрації слід нумерувати арабськими цифрами порядковою нумерацією в межах розділу, за винятком ілюстрацій, наведених у додатках. Номер ілюстрації складається з номера розділу і порядкового номера рисунка, відокремлених крапкою. Наприклад: «Рисунок 3.2» – це другий рисунок третього розділу.

Відстань між текстом і рисунком, рисунком та наступним текстом має бути один рядок: шрифт Times New Roman, розмір шрифта – 14, міжрядковий інтервал – 1,5. Відступ між рисунком і назвою рисунка не роблять.

Якщо рисунок не вміщується на одній сторінці, можна переносити його на інші сторінки, вміщуючи назву рисунка на першій сторінці, пояснювальні дані – на кожній сторінці, і під ними позначають: «Рисунок __, аркуш __».

Приклад підпису рисунка

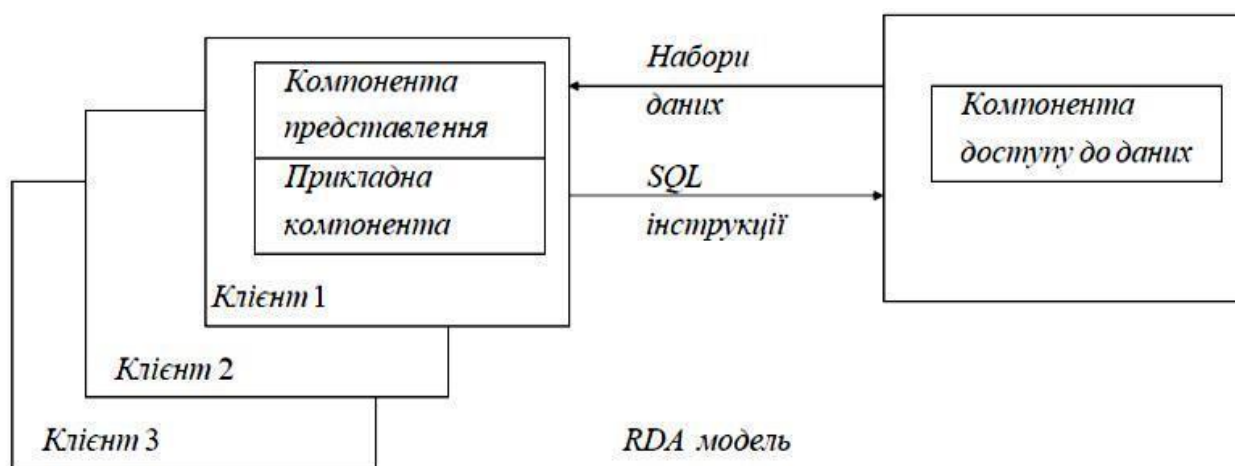


Рисунок 1.2 - Модель віддаленого доступу до даних

Таблиці потрібно нумерувати арабськими цифрами порядковою нумерацією в межах розділу (за винятком додатків). Номер таблиці складається з номера розділу й порядкового номера таблиці, відокремлених крапкою. Крапку після номера таблиці не ставлять. Наприклад: друга таблиця третього розділу позначається як «Таблиця 2.1»

Таблиця. 1.1

	Назва таблиці					
Головка						Заголовок граф
						Підзаголовки граф
Рядки						
	Боковик (заголовки рядків)		Графи колонок			

Кожна таблиця повинна мати назву, яку розміщують над таблицею і друкують симетрично до тексту. Назва має бути стислою і відображати зміст таблиці. Слово «Таблиця» та назву таблиці починають з великої літери. Назву не підкреслюють. Крапку після назви таблиці не ставлять.

Відстань між текстом і назвою таблиці, таблицею та наступним текстом має становити один рядок (Times New Roman, 14, міжрядковий інтервал – 1,5). Відступ між назвою таблиці і таблицею не роблять.

Заголовки таблиці починають з великої літери. У кінці заголовків крапки не ставлять. Заголовки граф зазначають в однині.

Допускається перенесення таблиці з великою кількістю рядків на іншу сторінку. При перенесенні назву таблиці вміщують тільки над першою частиною. Таблицю з великою кількістю граф допускається ділити на частини та розміщувати одну частину під іншою в межах однієї сторінки. Якщо рядки або графи таблиці виходять за формат сторінки, то в першому випадку в кожній частині таблиці повторюють її головку, у другому – боковик.

На всі таблиці повинні бути посилання в тексті, при цьому слово «таблиця» пишуть скорочено, наприклад: «... у табл. 1.1». У повторних посиланнях вказують скорочено слово «див.», наприклад: «див. табл. 1.3».

Формули та рівняння. Формули слід оформлювати відповідно до техніко-орфографічних правил. Великі та громіздкі формули, що містять знаки суми, добутку, диференціювання або інтегрування, а також усі нумеровані формули, розміщують на окремих рядках. Для економії місця кілька коротких однотипних формул можна записувати в одному рядку. Невеликі та нескладні формули, що не мають самостійного значення, допускається вписувати у текст.

Пояснення символів і коефіцієнтів подають під формулою у порядку їх появи, кожне – з нового рядка. Перший рядок пояснення починають зі слова «де» без двокрапки та без відступу.

Якщо рівняння не вміщується в один рядок, його переносять після знаків «=», «+», «-», «x» або «:» . Нумерують лише ті формули, на які є посилання у тексті, використовуючи арабські цифри в круглих дужках праворуч.

Номер формули складається з номера розділу та її порядкового номера, відокремлених крапкою (наприклад, «(1.7)»). Крапку після номера формули не ставлять. Для формул-дробів номер подають на рівні основної горизонтальної риски формули.

Формула є частиною речення, тому перед нею та після неї розділові знаки ставлять відповідно до правил пунктуації. Двокрапку перед формулою застосовують лише у випадках, передбачених правилами пунктуації.

Посилання на формули подають порядковим номером формули в дужках, наприклад: «... у формулі (2.1)».

Додатки оформляють як продовження роботи на наступних сторінках, розташовуючи їх відповідно до появи посилань у тексті. Додатки повинні мати спільну з основним текстом наскрізну нумерацію сторінок.

У додатках розміщують матеріал, який не може бути послідовно наведений в основній частині через великий обсяг, суто технічний характер або специфіку відтворення (наприклад, матеріали іншого формату). У додатки можуть бути внесені окремі ілюстрації, таблиці, схеми тощо.

У додатках текстової частини курсового проєкту з ООП С# обов'язково подаються: «Технічне завдання» та «Керівництво користувача» (інструкція користувача).

Ілюстрації, таблиці, формули та рівняння, наведені у додатку, нумерують у межах кожного додатку, наприклад: «Рисунок А.3», «Таблиця А.2», формула (А.1).

Марковані списки (переліки), за потреби, можуть бути наведені всередині пунктів або підпунктів. Перед переліком ставлять двокрапку.

Перед кожною позицією переліку слід ставити малу літеру української абетки з дужкою, або, не нумеруючи – дефіс (першій рівень деталізації).

Для подальшої деталізації переліку слід використовувати арабські цифри з дужкою (другий рівень деталізації).

Переліки першого рівня деталізації друкують малими літерами з абзацного відступу: положення маркера на відстані 1,25 см від лівої межі сторінки, положення першого рядка переліку – 1,75 см від лівої межі сторінки, наступні рядки поточної позиції переліку (за наявності) друкують з нульовим відступом від лівої межі сторінки. Переліки другого рівня друкують малими літерами з відступом відносно місця розташування переліків першого рівня: положення маркера на відстані 1,75 см від лівої межі сторінки, положення першого рядка переліку – 2,25 см від лівої межі сторінки, наступні рядки позиції переліку (за наявності) друкують з відступом 1,25 см від лівої межі сторінки.

Правила оформлення списку використаних джерел наведено в додатку 3. Список використаних джерел оформлюється відповідно до Національного стандарту України ДСТУ 8302:2015.

Загальні вимоги до оформлення списку використаних джерел:

- список формують за алфавітним порядком; джерела латиницею розміщують після джерел кирилицею;
- кожне джерело включають у список лише один раз (незалежно від кількості посилань у тексті);
- бібліографічний опис подають мовою оригіналу (не перекладають);
- список оформлюють нумерованим переліком: після номера ставлять крапку; відступ від номера до тексту – 0,5 см; другий та наступні рядки оформлюють з відступом 0,75 см від лівого поля.

Джерела можна розмішувати одним із таких способів: у порядку появи посилань у тексті (найбільш зручний для користування і рекомендований при написанні наукових робіт), в алфавітному порядку прізвищ перших авторів або заголовків.

Рекомендована кількість використаних джерел — не менше 25.

2.4. Вимоги до структури та оформлення презентації

Презентація є важливою частиною захисту курсового проєкту, що впливає на оцінювання та повинна бути правильно оформлена.

Основні цілі презентації:

- продемонструвати здатність студента аналізувати, систематизувати та доступно викладати матеріал;
- розвинути ораторські та комунікаційні навички під час доповіді, відповідей на запитання та участі в обговоренні;

- сформувати вміння працювати з програмним забезпеченням для створення та демонстрації презентацій з урахуванням специфіки дисципліни «ООП С#».

Вимоги до змісту презентації:

- відповідність змісту презентації поставленим цілям і завданням курсового проєкту;
- дотримання правил орфографії, пунктуації, скорочень і загальних правил оформлення тексту (зокрема, відсутність крапки в заголовках);
- відсутність фактичних помилок, достовірність представленої інформації;
- лаконічність тексту на слайді (не більше 40 слів, не більше 3-4 речень);
- логічна завершеність (кожен смисловий фрагмент має бути завершеним);
- об'єднання семантично пов'язаних елементів у цілісні блоки;
- стислість та інформативність;
- коректне розташування інформації на слайді (найбільш важлива інформація – у центральній частині; підписи до зображень розташовуються під ними; форматування тексту за шириною; не допускаються «рвані» краї тексту);
- наявність не більше одного логічного наголосу на слайді (виділення кольором, рамкою тощо).

Вимоги до оформлення презентації:

- використання єдиного стилю оформлення;
- відповідність стилю презентації (графічного, звукового, анімаційного) її змісту;
- використання для фону психологічно комфортних тонів (рекомендовано – світлий фон, зокрема білий);
- фон повинен бути елементом заднього плану та не погіршувати читабельність інформації;
- використання не більше трьох кольорів на одному слайді (фон, заголовки, основний текст);
- доцільність використання анімаційних ефектів (без надлишкових ефектів);
- читабельність тексту на тлі слайда (контрастність фону і тексту);
- використання шрифтів без засічок та не більше 1–2 варіантів шрифту в межах презентації;
- рекомендована довжина рядка не більше 36 знаків;
- міжрядковий інтервал всередині абзацу – 1,5; між абзацами – 2 інтервали;
- підкреслення застосовується лише для гіперпосилань.

Нижче наведено рекомендовану структуру презентації для захисту курсового проєкту з ООП С#.

Слайд 1. Тема, виконавець, науковий керівник.

Слайд 2. Обґрунтування актуальності (1-2 речення).

Слайд 3. Мета роботи.

Слайд 4. Завдання роботи.

Слайд 5. Об'єкт і предмет дослідження (за потреби).

Слайд 6. Основні поняття та терміни, що використовуються в роботі.

Слайд 7. Основні підходи/алгоритми або методи, що були проаналізовані у першому розділі.

Слайд 8. Аналогічні програмні продукти (2–3 приклади): скріншоти, назви, короткі висновки (переваги/недоліки).

Слайд 9. Призначення програмного продукту та вимоги до нього (функціональні вимоги).

Слайд 10. Загальна структура проєкту (компоненти, шари, модулі)

Слайд 11. UML/ООП-модель (діаграма класів, ключові абстракції, взаємодія класів).

Слайд 12. Інструментальні засоби розробки (.NET, IDE, NuGet, Git).

Слайд 13-16. Основні режими роботи.

Слайд 17. Висновки.

Слайд 18. Дякую за увагу!

Дана структура є шаблоном для формування презентації і носить рекомендаційний характер.

Під час підготовки презентації кількість слайдів може змінюватися, але рекомендовано дотримуватись наведеного порядку викладу.

2.5. Вимоги до відеоролика-представлення роботи

Одним із обов'язкових компонентів курсового проєкту є відеоролик-презентація, що демонструє роботу розробленого програмного продукту. Відеоролик має висвітлювати основні можливості програмної реалізації та відповідати таким вимогам:

- тривалість від 1 до 5 хвилин;
- якість відео: не нижче 720p, частота кадрів – не менше 30 fps;
- формат відеофайлу: контейнер .mkv (або .mp4), кодек H.264, співвідношення сторін — 16:9;
- допускається монтаж (відео може бути знято не одним дублем);
- титри (за наявності) повинні бути вмонтовані у відеоряд (не окремим файлом).

Вимоги до структури відеоролика:

- відомості про автора (ПІБ), назву програмного продукту та тему курсового проєкту;
- демонстрацію основних сценаріїв роботи з програмним продуктом (основний функціонал);
- у разі необхідності інсталяції програмного продукту — демонстрацію процесу встановлення або запуску (інструкція запуску);
- коментарі до демонстрації (у вигляді текстових пояснень на екрані та/або голосового супроводу автора).

2.6. Підготовка до виступу-представлення наукової роботи

Перед захистом курсового проєкту здобувач вищої освіти повинен підготувати доповідь та представити її на засіданні комісії.

Тривалість доповіді має становити 7–10 хвилин. Виступ супроводжується презентацією, оформленою відповідно до встановлених вимог.

У доповіді необхідно:

- обґрунтувати актуальність теми;
- визначити об'єкт і предмет дослідження (за потреби);
- сформулювати мету та завдання курсового проєкту;
- стисло висвітлити зміст кожного розділу курсового проєкту, акцентуючи увагу на ключових рішеннях та отриманих результатах;
- описати основні функціональні можливості розробленого програмного продукту та особливості реалізації (архітектура, ООП-модель, інструменти);
- сформулювати висновки, а також підкреслити практичне значення отриманих результатів і можливості подальшого розвитку програмного продукту.

Після завершення доповіді здобувач вищої освіти повинен бути готовий відповісти на запитання членів комісії та за необхідності продемонструвати роботу програмного продукту (або відеоролик-презентацію).

2.7. Апробація отриманих результатів

Апробація результатів є одним із видів науково-практичної діяльності, що передбачає перевірку отриманих результатів дослідження або розробки шляхом їх представлення, обговорення та/або практичного застосування. Метою апробації є визначення рівня обґрунтованості отриманих результатів, їх практичної цінності та можливості використання для розв'язання прикладних задач.

Апробація результатів курсового проєкту є бажаною та може здійснюватися за наявності відповідних можливостей (у тому числі в межах кафедри, факультету, освітніх або наукових заходів).

Форми апробації:

- участь у бесідах, засіданнях або круглих столах на кафедрі з обговоренням результатів виконаної роботи;
- представлення доповідей або демонстрація програмної розробки під час проведення Днів науки факультету (університету);
- підготовка та публікація матеріалів за темою дослідження (статті, тези, доповіді) у виданнях або збірниках конференцій;
- участь у наукових конференціях, семінарах, студентських конкурсах, хакатонах (за тематикою розробки);
- оприлюднення результатів роботи в межах практичної діяльності підприємства, організації або установи (за наявності співпраці);

- впровадження запропонованих рішень у діяльність підприємств, установ або організацій (пілотне використання, тестування);
- використання результатів під час підготовки навчально-методичних матеріалів (практичних занять, лабораторних робіт, лекцій, інформаційного забезпечення дисциплін).

2.8. Академічна доброчесність при написанні курсового проєкту

Дотримання здобувачами вищої освіти принципів академічної доброчесності під час виконання курсового проєкту з ООП С# регламентується ст. 42 Закону України «Про освіту», Кодексом академічної доброчесності ДУІКТ, а також Положенням про систему запобігання та виявлення академічного плагіату у науково-дослідній діяльності здобувачів вищої освіти і науково-педагогічних працівників ДУІКТ.

У разі виявлення науковим керівником у роботі здобувача одного або кількох порушень академічної доброчесності, зокрема: академічного плагіату, самоплагіату, фабрикації, фальсифікації, обману, до здобувача можуть бути застосовані заходи відповідальності, передбачені Кодексом академічної доброчесності ДУІКТ. До таких заходів можуть належати, зокрема:

- повторне проходження оцінювання;
- повторне проходження відповідного освітнього компонента освітньої програми;
- відрахування з університету;
- позбавлення академічної стипендії;
- позбавлення наданих університетом пільг з оплати навчання.

У випадку, якщо порушення академічної доброчесності виявлені не менш ніж за 3–4 тижні до захисту курсового проєкту, здобувачу може бути надана можливість виправлення виявлених недоліків (усунення порушень, коректне оформлення посилань тощо). Якщо порушення виявлені менш ніж за 2 тижні до захисту, курсовий проєкт не допускається до захисту, здобувач отримує оцінку «незадовільно» з можливістю повторного захисту відповідно до встановленого порядку.

3. Порядок захисту й оцінювання курсового проєкту

3.1. Процедура захисту

Допуск здобувача вищої освіти до захисту курсового проєкту здійснює науковий керівник. Критеріями допуску є:

- наявність електронного варіанту текстової частини курсового проєкту у форматі .doc або .docx, оформленого згідно з встановленими вимогами;
- наявність електронного варіанту працюючої програмної розробки, що відповідає поставленій у роботі задачі, у вигляді вихідного коду проєкту (рішення .sln, проєкт .csproj) та/або зібраного застосунку;

- наявність інструкції запуску та використання програмного продукту (README/керівництво користувача), де зазначено вимоги до середовища (версія .NET, залежності тощо);
- наявність посилання на Git-репозиторій (за наявності/вимог кафедри), що містить актуальну версію проєкту та історію виконання роботи;
- наявність переплетеного друкованого примірника текстової частини курсового проєкту, оформленого згідно з вимогами та завізованого керівником;
- відповідність змісту пояснювальної записки темі курсового проєкту;
- наявність у додатках технічного завдання та керівництва користувача (інструкції користувача);
- наявність відеоролика-представлення програмної розробки;
- дотримання принципів академічної доброчесності під час виконання курсового проєкту відповідно до чинних нормативних документів.

Захист курсового проєкту проводиться перед комісією, до складу якої входять щонайменше два викладачі кафедри та керівник курсового проєкту. Дата проведення захисту визначається відповідно до графіка підсумкового семестрового контролю факультету.

Процедура захисту включає коротку доповідь здобувача вищої освіти з використанням презентації, відповіді на запитання членів комісії, а також (за потреби) демонстрацію роботи програмного продукту або відеоролика. У доповіді слід висвітлити актуальність теми, мету та завдання курсового проєкту, отримані результати, а також особливості реалізації програмного продукту. Крім того, здобувач вищої освіти повинен продемонструвати здатність аргументовано відповідати на запитання з предметної області роботи та брати участь в обговоренні.

Після завершення захисту комісія приймає рішення щодо підсумкової оцінки, враховуючи встановлені критерії оцінювання. Результати захисту оголошуються здобувачам вищої освіти у той же день.

Отримана оцінка вноситься до заліково-екзаменаційної відомості, індивідуального навчального плану та залікової книжки здобувача вищої освіти, засвідчується підписами членів комісії та враховується під час визначення розміру стипендії разом з іншими підсумковими оцінками.

У разі отримання підсумкової сумарної оцінки менше 60 балів за 100-бальною шкалою або у випадку недопуску курсового проєкту до захисту у заліково-екзаменаційній відомості робиться відповідний запис про академічну заборгованість.

Здобувач вищої освіти не допускається до захисту курсового проєкту у випадках:

- невиконання критеріїв допуску;
- порушення термінів подання роботи на кафедру без поважних причин;
- виявлення порушень академічної доброчесності.

Ліквідація академічної заборгованості здійснюється шляхом повторного виконання та захисту курсового проєкту. У випадку грубих порушень академічної доброчесності курсовий проєкт виконується повторно за новою темою. В інших випадках повторний захист допускається після усунення недоліків у поданій роботі. Допуск до повторного захисту здійснюється у встановлений термін ліквідації академічної заборгованості.

Інші випадки (хвороба, відрядження тощо) регламентуються Положенням про організацію навчального процесу.

3.2. Критерії оцінювання курсового проєкту

Оцінки за якість виконання курсового проєкту та результати її захисту відображаються в сумарній підсумковій оцінці і виставляються за спільної згоди членами комісії.

Критерії оцінювання на захисті комісією:

№	Вид роботи	Кількість Балів
1	Обґрунтування актуальності теми	0-5
2	Відповідність змісту текстової частини темі курсового проєкту	0-10
3	Повнота розкриття проблеми та якість теоретичного аналізу (огляд літератури, аналіз аналогів, вимоги до системи)	0-20
4	Складність і якість реалізації програмного продукту (ООП-модель, архітектура, взаємодія класів, використання C#/.NET можливостей) та особистий внесок здобувача	0-30
5	Відеоролик-презентація програмної розробки	0-5
6	Презентація доповіді	0-5
7	Апробація результатів курсового проєкту	0-5
8	Захист курсового проєкту: доповідь та чіткі, обґрунтовані відповіді на запитання	0-20

Шкала оцінювання (національна та ECTS)

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
90 – 100	A	Відмінно
82–89	B	Добре
75-81	C	
67 -74	D	Задовільно
60-66	E	
1–59	FX	Незадовільно

Список використаних джерел

1. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Boston : Pearson, 2017. 432 p.
2. Martin R. C. Agile Software Development: Principles, Patterns, and Practices. Upper Saddle River : Prentice Hall, 2002. 552 p.
3. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston : Addison-Wesley, 1994. 395 p.
4. Fowler M. Refactoring: Improving the Design of Existing Code. 2nd ed. Boston : Addison-Wesley, 2018. 448 p.
5. Evans E. Domain-Driven Design: Tackling Complexity in the Heart of Software. Boston : Addison-Wesley, 2003. 560 p.
6. McConnell S. Code Complete: A Practical Handbook of Software Construction. 2nd ed. Redmond : Microsoft Press, 2004. 960 p.
7. Richter J. CLR via C#. 4th ed. Redmond : Microsoft Press, 2012. 896 p.
8. Skeet J. C# in Depth. 4th ed. Shelter Island : Manning Publications, 2019. 528 p.
9. Albahari J., Albahari B. C# 12 in a Nutshell. Sebastopol : O'Reilly Media, 2024. 1056 p.
10. Troelsen A., Japikse P. Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming. Berkeley : Apress, 2022. 1750 p.
11. Microsoft Learn. C# documentation. URL: <https://learn.microsoft.com/dotnet/csharp/> (дата звернення: 15.01.2026).
12. Microsoft Learn. .NET documentation. URL: <https://learn.microsoft.com/dotnet/> (дата звернення: 15.01.2026).
13. Microsoft Learn. ASP.NET Core documentation. URL: <https://learn.microsoft.com/aspnet/core/> (дата звернення: 15.01.2026).
14. Microsoft Learn. Entity Framework Core documentation. URL: <https://learn.microsoft.com/ef/core/> (дата звернення: 15.01.2026).
15. Microsoft Learn. Dependency injection in .NET. URL: <https://learn.microsoft.com/dotnet/core/extensions/dependency-injection> (дата звернення: 15.01.2026).
16. GitHub Docs. About repositories. URL: <https://docs.github.com/> (дата звернення: 15.01.2026).
17. NuGet Documentation. URL: <https://learn.microsoft.com/nuget/> (дата звернення: 15.01.2026).
18. xUnit.net documentation. URL: <https://xunit.net/> (дата звернення: 15.01.2026).
19. NUnit documentation. URL: <https://nunit.org/> (дата звернення: 15.01.2026).
20. FluentValidation documentation. URL: <https://docs.fluentvalidation.net/> (дата звернення: 15.01.2026).
21. ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Geneva : ISO, 2011.
22. ISO/IEC 12207:2017. Systems and software engineering — Software life cycle processes. Geneva : ISO, 2017.

23. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання: Загальні положення та правила складання. Київ : ДП «УкрНДНЦ», 2015. 16 с.
24. ДСТУ 3582:2013. Бібліографічний опис. Скорочення слів і словосполучень українською мовою. Загальні вимоги та правила. Київ : Мінекономрозвитку України, 2014. 15 с.
25. Про освіту : Закон України від 05.09.2017 р. № 2145-VIII. Голос України. 2017. № 178–179.
26. Про вищу освіту : Закон України від 01.07.2014 р. № 1556-VII. URL: <https://zakon.rada.gov.ua/laws/show/1556-18> (дата звернення: 15.01.2026).

Зразок титульної сторінки курсового проєкту з ХХХ

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра Технології цифрового розвитку

Курсовий проєкт з ХХХ

**РЕАЛІЗАЦІЯ АЛГОРИТМУ РОЗКЛАДАННЯ ГРАФУ ЗА ЙОГО
ВЕРШИНАМИ**

Виконав:
ХХХ,
студент групи ТЦР-ХХ
факультету ХХХ

Науковий керівник:
Іванчук Іван Іванович,
кандидат
фіз.-мат наук, доцент кафедри
інженерії програмного забезпечення

Київ 2025

ЗМІСТ

ВСТУП

РОЗДІЛ 1. НАЗВА РОЗДІЛУ ...

1.1. Назва ...

1.1.1. Назва ...

1.2. Назва ...

1.2.1. Назва ...

1.2.2. Назва ...

1.3. Назва ...

1.4. Огляд та аналіз аналогічних програмних розробок

РОЗДІЛ 2. НАЗВА РОЗДІЛУ ...

2.1. Постановка задачі, призначення та вимоги до програмного засобу
«назва програмної розробки»

2.2. Загальний опис проєкту

2.3. Обґрунтування вибору інструментальних засобів розробки *«назва програмної розробки»*

2.4. Особливості програмної реалізації *«назва програмної розробки»*

2.5. Організація тестування та налагодження програмного засобу *«назва програмної розробки»*

2.6. Рекомендації щодо використання та впровадження програмного засобу *«назва програмної розробки»*

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Додатки

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ОГЛЯД ЗАДАЧ РОЗКЛАДАННЯ ГРАФУ ЗА ЙОГО ВЕРШИНАМИ	4
1.1. Поняття графу, Способи представлення графів в пам'яті комп'ютера.....	4
1.2. Формулювання задачі розкладання графу за його вершинами. Поняття хроматичного числа графу.....	6
1.3. Основні алгоритми розкладання графів за його вершинами.....	8
1.3.1. Розкладання графу за його кістяками.....	8
1.3.2. Послідовний алгоритм розкладання.....	10
1.3.3. Точний алгоритм розкладання	12
1.4. Порівняльний аналіз алгоритмів розкладання графу за його вершинами.....	14
1.5. Огляд та аналіз програмних засобів, що реалізують задачу розкладання графу за його вершинами.....	15
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ “COLORGRAPH” ДЛЯ РОЗКЛАДАННЯ ГРАФУ ЗА ЙОГО ВЕРШИНАМИ ..	17
2.1. Постановка задачі, призначення та вимоги до програмного засобу “COLORGRAPH”	17
2.2. Опис алгоритму розкладання графу за його кістяками.....	18
2.3. Обґрунтування вибору інструментальних засобів розробки програмного засобу “COLORGRAPH”.....	20
2.4. Особливості програмної реалізації	22
2.5. Основні режими функціонування програмного засобу “COLORGRAPH”	24
2.6. Рекомендації з використання та впровадження програмного засобу.....	26
ВИСНОВКИ.....	28
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	30
Додатки.....	32

Unified system for program documentation.

Technical specification for development.

Даний стандарт встановлює порядок побудови і оформлення технічного завдання на розробку програми чи програмного продукту для персональних комп'ютерів (ПК), програмних комплексів і систем незалежно від їх призначення та сфери застосування.

Загальні положення.

1.1. Технічне завдання оформляють за допомогою комп'ютера на одній стороні аркуша білого паперу формату А4 (210×297 мм) через півтора міжрядкових інтервали (текстовий редактор Microsoft Word, шрифт Times New Roman, розмір 14).

Обсяг технічного завдання повинен становити 3–4 сторінки основного тексту.

Текст документа необхідно друкувати, залишаючи береги таких розмірів: лівий – 20 мм, правий – 10 мм, верхній – 20 мм, нижній – 20 мм.

Структурними елементами технічного завдання є розділи.

Заголовки розділів пишуть прописними буквами і розміщують симетрично відносно правої і лівої меж тексту (Times New Roman, 14, напівжирний).

Технічне завдання повинно містити такі розділи:

вступ;

підстави для розробки;

призначення розробки;

вимоги до програми чи програмного продукту;

вимоги до програмної документації;

техніко-економічні показники;

стадії і етапи розробки;

порядок контролю і приймання;

в технічне завдання допускається включати додатки.

Залежно від особливостей програми, її специфіки допускається уточнити зміст розділів, вводити нові розділи чи об'єднати окремі з них.

2. Зміст розділів

2.1. У розділі «Вступ» вказують найменування програмного продукту, коротку характеристику галузі застосування, а також опис об'єкта/процесу, в якому використовують програму чи програмний продукт.

2.2. У розділі «Підстави для розробки» повинні бути зазначені: документ(и), на підставі яких здійснюється розробка; організація, що затвердила цей документ, та дата затвердження; найменування і/або умовне позначення теми розробки.

2.3. У розділі «Призначення розробки» зазначають функціональне і експлуатаційне призначення програмного продукту.

2.4. Розділ «Вимоги до програми чи програмного продукту» повинен містити такі підрозділи:

- вимоги до функціональних характеристик;
- вимоги до надійності; умови експлуатації;
- вимоги до складу і параметрів технічних засобів;
- вимоги до інформаційної і програмної сумісності;
- вимоги до маркування і упаковки; вимоги до транспортування і збереження; спеціальні вимоги.

2.4.1. В підрозділі «Вимоги до функціональних характеристик» повинні бути вказані вимоги до складу виконуваних функцій, організації вхідних і вихідних даних, часових характеристик тощо.

2.4.2. В підрозділі «Вимоги до надійності» повинні бути вказані вимоги до забезпечення стійкого функціонування: контроль вхідних даних, обробка помилок, відновлення після збоїв тощо.

2.4.3. В підрозділі «Умови експлуатації» повинні бути вказані умови експлуатації, при яких повинні забезпечуватись задані характеристики, а також необхідна кількість і кваліфікація персоналу (користувачів).

2.4.4. В підрозділі «Вимоги до складу і параметрів технічних засобів» вказують необхідний склад технічних засобів із зазначенням їх основних характеристик.

2.4.5. В підрозділі «Вимоги до інформаційної і програмної сумісності» повинні бути визначені вимоги до інформаційних структур на вході і виході, методів розв'язування задач, мов програмування і програмних засобів, що використовуються програмою.

Для програмної реалізації на платформі C#/NET у технічному завданні доцільно зазначати:

- версію платформи .NET (Runtime/SDK);
- підтримувані операційні системи;
- використані сторонні бібліотеки (NuGet-пакети);
- вимоги до СКБД (за наявності);
- вимоги до захисту інформації (за потреби).

2.4.6. В підрозділі «Вимоги до маркування і упаковки» в загальному випадку вказують вимоги до маркування програмного виробу, варіанти і способи упаковки.

2.4.7. В підрозділі «Вимоги до транспортування і збереження» повинні бути вказані для програмного виробу умови транспортування, місце збереження та ін.

2.5. В розділі «Вимоги до програмної документації» повинен бути вказаний попередній склад програмної документації і, за необхідності, спеціальні вимоги до неї.

2.6. В розділі «Техніко-економічні показники» повинні бути вказані: орієнтовна економічна ефективність, передбачувана річна потреба, економічні

переваги розробки у порівнянні з кращими вітчизняними і зарубіжними зразками чи аналогами.

2.7. В розділі «Стадії і етапи розробки» встановлюють стадії розробки, етапи і зміст робіт, перелік програмних документів, терміни виконання та визначають виконавців.

2.8. В розділі «Порядок контролю і приймання» повинні бути вказані види випробувань і загальні вимоги до приймання програмного продукту.

2.9. У додатках до технічного завдання, за необхідності, подають:
перелік науково-дослідних робіт та інших матеріалів, що обумовлюють розробку;
схеми алгоритмів, таблиці, описи, обґрунтування, розрахунки;
інші матеріали та джерела, що можуть бути використані при розробці.

Інструкція користувачу**1. Загальні відомості**

У даному розділі зазначаються:

- повна назва програмного продукту;
- версія (за наявності);
- автор (ПІБ, група);
- платформа реалізації: C# / .NET;
- тип застосунку (консольний, десктопний, Web API тощо).

2. Функціональне призначення

Наводяться відомості про призначення програмного продукту та перелік задач, які можуть бути розв'язані за допомогою даної програмної розробки. Коротко описується, для кого призначений продукт та які основні функції реалізовано.

3. Умови застосування програми

Вказуються умови, необхідні для коректної роботи програмного продукту:

- тип пристрою (ПК, ноутбук);
- операційна система (Windows, Linux, macOS – залежно від реалізації);
- мінімальні характеристики: процесор, оперативна пам'ять, вільне місце на диску;
- необхідне програмне забезпечення: .NET Runtime (або .NET SDK, якщо запуск через командний рядок), IDE (за потреби для запуску з коду: Visual Studio, Rider, VS Code), додаткові компоненти (SQL Server, PostgreSQL, SQLite тощо – за потреби);
- вимоги до підключення до Інтернету (за потреби).

4. Для кожного повідомлення вказують:

Описуються кроки запуску програмного продукту залежно від типу реалізації:

- запуск готового виконуваного файлу (якщо є);
- запуск проєкту з вихідного коду: відкриття рішення (.sln), відновлення залежностей (NuGet restore), запуск проєкту;
- опис початкових налаштувань (конфігураційні файли, connection string тощо).

5. Повідомлення користувачу

Наводяться приклади можливих повідомлень програми:

- повідомлення про помилки;
- повідомлення про некоректні вхідні дані;
- попередження;
- інформаційні повідомлення.

Для кожного повідомлення вказують:

- текст повідомлення;
- причину виникнення;
- рекомендовані дії користувача.

6. Опис роботи програми

Описується функціонал програмного продукту та послідовність дій користувача:

- головні режими роботи;
- основні екрани/форми/команди;
- сценарії використання (кроки: 1, 2, 3);
- приклади введення даних і отримання результатів.

*Приклади
оформлення бібліографічного опису
у списку використаних джерел з урахуванням національного стандарту України
ДСТУ 8302:2015*

Характеристика джерела	Приклад оформлення джерела
Книги: Один автор	1. Колесников О. В. Основи наукових досліджень : навч. посіб. 2-ге вид., перероб. та доп. Київ : ЦУЛ, 2011. 144 с. 2. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Boston : Pearson, 2017. 432 p. 3. Fowler M. Refactoring: Improving the Design of Existing Code. 2nd ed. Boston : Addison-Wesley, 2018. 448 p.
Два або три автори	1. Месюра В. І., Лисак Н. В., Суприган О. І. Математичні основи логічного програмування : навч. посіб. Вінниця : ВНТУ, 2013. 93 с. 2. Albahari J., Albahari B. C# 12 in a Nutshell. Sebastopol : O'Reilly Media, 2024. 1056 p. 3. Freeman E., Robson E. Head First Design Patterns. 2nd ed. Sebastopol : O'Reilly Media, 2021. 694 p.
Чотири і більше авторів	1. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston : Addison-Wesley, 1994. 395 p. 2. Операційне числення : навч. посіб. / С. М. Гребенюк та ін. Запоріжжя : ЗНУ, 2015. 88 с. 3. Основи охорони праці : підручник / О. І. Запорожець та ін. 2-ге вид. Київ : ЦУЛ, 2016. 264 с.
Автор(и) та редактор(и) / упорядники	1. Енциклопедія Сучасної України / редкол.: І. М. Дзюба та ін. Київ : САМ, 2016. Т. 17. 712 с. 2. Каталог наукових видань співробітників НУВГП 2014–2019 / упоряд.: Є. Г. Герасімов, О. Л. Пінчук, В. М. Чабан. Рівне : НУВГП, 2020. 116 с. 3. ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE). Geneva : ISO, 2011.
Багатотомні видання	1. Knuth D. E. The Art of Computer Programming : in 4 vols. Boston : Addison-Wesley, 1997. Vol. 1. 650 p.

	2. Правова система України: історія, стан та перспективи : у 5 т. / Акад. прав. наук України. Харків : Право, 2009. Т. 2. 576 с.
Автореферати дисертацій	1. Савенко О. С. Теорія та практика створення розподілених систем виявлення зловмисного програмного забезпечення в локальних комп'ютерних мережах : автореф. дис. ... д-ра техн. наук : 05.13.06. Львів, 2019. 43 с. 2. Author S. Dissertation title : abstract of dissertation ... PhD. University, City, Year. 45 p.
Дисертації	1. Левчук С. А. Матриці Гріна рівнянь і систем еліптичного типу для дослідження статичного деформування складених тіл : дис. ... канд. фіз.-мат. наук : 01.02.04. Запоріжжя, 2002. 150 с. 2. Author S. Thesis title : PhD thesis. University, City, Year. 180 p.
Законодавчі та нормативні документи	1. Про освіту : Закон України від 05.09.2017 р. № 2145-VIII. Голос України. 2017. 27 верес. (№ 178–179). С. 10–22. 2. Про вищу освіту : Закон України від 01.07.2014 р. № 1556-VII. URL: https://zakon.rada.gov.ua/laws/show/1556-18 (дата звернення: 15.01.2026). 3. Про затвердження Вимог до оформлення дисертації : наказ МОН України від 12.01.2017 р. № 40. Офіційний вісник України. 2017. № 20. С. 136–141.
Архівні документи	1. Лист Голови Спілки «Чорнобиль» Г. Ф. Лепіна на ім'я Голови Ради Міністрів УРСР В. А. Масола щодо реєстрації Статуту Спілки та сторінки Статуту. 14 грудня 1989 р. ЦДАГО України. Ф. 1. Оп. 32. Спр. 2612. Арк. 63, 64 зв., 71. 2. Наукове товариство ім. Шевченка. Львів. Наук. б-ка ім. В. Стефаника НАН України. Ф. 1. Оп. 1. Спр. 78. Арк. 1–7.
Патенти	1. Люмінісцентний матеріал : пат. 25742 Україна : МПК6 C09K11/00, G01T1/28, G21H3/00. № 200701472; заявл. 12.02.2007; опубл. 27.08.2007, Бюл. № 13. 4 с. 2. Спосіб лікування синдрому дефіциту уваги та гіперактивності у дітей : пат. 76509 Україна. № 2004042416; заявл. 01.04.2004; опубл. 01.08.2006, Бюл. № 8 (кн. 1). 120 с.

Стандарти	1. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання: Загальні положення та правила складання. [Чинний від 2016-07-01]. Київ : ДП «УкрНДНЦ», 2015. 16 с. 2. ISO/IEC 12207:2017 Systems and software engineering – Software life cycle processes. Geneva : ISO, 2017. 3. ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE). Geneva : ISO, 2011.
Частина видання: матеріали конференцій (тези, доповіді)	1. Пруц Г. С. Особливості очистки системного розділу ОС Microsoft Windows 10. Математика. Інформаційні технології. Освіта : тези доп. VII міжн. наук.-практ. конф. (м. Луцьк, 3–5 червн. 2018 р.). Луцьк, 2018. С. 51–52. 2. Author S. Paper title. Proceedings of the Conference Name (City, Date). City, Year. P. 20–22.
Частина видання періодичного (журнал, газета)	1. Гнатченко Д. А. Системи автоматизованого проектування як інноваційний напрям розвитку. Комп'ютерно-інтегровані технології: освіта, наука, виробництво. 2020. № 40. С. 10–15. 2. Yoshida T., Oka T., Kitagawa M. Photoconductivity of CdS Crystals Irradiated with fast Electrons. Applied Physics Letters. 1972. Vol. 20, No. 1. P. 1–2.
Електронні ресурси	1. Microsoft Learn. C# documentation. URL: https://learn.microsoft.com/dotnet/csharp/ (дата звернення: 15.01.2026). 2. Microsoft Learn. ASP.NET Core documentation. URL: https://learn.microsoft.com/aspnet/core/ (дата звернення: 15.01.2026). 3. Microsoft Learn. Entity Framework Core documentation. URL: https://learn.microsoft.com/ef/core/ (дата звернення: 15.01.2026). 4. GitHub Docs. URL: https://docs.github.com/ (дата звернення: 15.01.2026). 5. NuGet documentation. URL: https://learn.microsoft.com/nuget/ (дата звернення: 15.01.2026).